REMARKS

In the Office Action mailed March 26, 2007, the examiner rejected claims 27, 28, 32, 35-38, 40, 42-44, 46, 48 and 49 on the ground of anticipation by Altberg.

Altberg discloses "a method for providing the ability to install required files that are necessary for an application's execution without prompting a user to perform a separate installation procedure" (Altberg, column 3, lines 26-30). Altberg refers to installing an application onto a computer system by copying files from a source location such as an installation disk (see column 1, lines 12-20). Therefore, the "installation" referred to in Altberg means the installation of an application on a computer system for the first time, for instance, when an application is first installed from a CD-ROM onto the hard disk of a computer so that the application may be run from the computer.

The files which are installed onto a computer system using the method of Altberg are stored in predetermined positions and the location of these positions are known by the application 205, a shared library DLL 210 and an installer module 220 (column 5, lines 36-38). Only once the application has been installed on the computer system the application can the application be "launched" and used. When the application 205 is launched the application calls the shared library DLL 210 to determine whether the required files for the application exist on the computer system (column 5, lines 49-51). If a required file is missing, the application 205 invokes the installer module 220 to replace the missing required files from copies which may be located in the computer system or may be located on separate shipping media (column 5, line 66 to column 6, line 3).

After the installer module 220 has installed missing required files, the application is re-initiated so that the application can be executed and utilise all of the required files which are now present (column 7, lines 36-40, column 8, lines 8-9).

Altberg describes how there can be multiple files that are required by an application and some of the required files may be used by other applications or other computer processes that are being executed in the computer system (column 8, lines 15-20). In this case, the installer module identifies if there are newer versions of the

required files than the versions which are present in the computer system as well as replacing any missing required files. In order to be sure that a file which is being replaced by a newer version by the installer is not in use by another application in the computer system the entire computer system must be restarted before the file can be replaced and the application can be launched (column 8, lines 25-28).

In contrast to Altberg, the present invention incorporates a single executable application that contains a program, one or more encrypted sub-routines and a decryption routine. The present application describes how the application is "installed" (see figures 3). When the present application refers to "installing" the application the present application means loading the application into RAM ready for execution. The "installation" of the application of the present invention occurs every time the application is launched. When the application of the present invention is launched the program is installed in RAM (i.e. the program is executed within the computer system). During execution, the program requires access to one or more sub-routines. When access to a sub-routine is required the decryption routine detects whether the sub-routine is already available within the computer system. If the sub-routine is not available, the decryption routine decrypts the required encrypted sub-routine so that the required sub-routine can be loaded into the RAM for access by the program.

The claims of the present application have been amended to highlight the differences between the present invention and Altberg, The independent claims now specify that the "decryption routine is operable during execution of the application to detect whether a required sub-routine is already available within the computer system to cause the program to use the sub-routine within the computer system if already available, and to decrypt the required encrypted sub-routine into an executable form if the sub-routine is not already available within the computer system".

Altberg fails to disclose or teach an executable which incorporates a decryption routine that is operable during execution of the application to detect and install required sub-routines if required. The method disclosed in Altberg is, by contrast, a first time installer which installs required files in a computer system so

that the required files are present when an application which requires the files is executed. Altberg utilises a separate installer module which is used to install required files before a separate application is executed. The "installing" which occurs in Altberg is the installation of files onto a computer system before the application is executed (i.e. before a program is launched and loaded into RAM).

In the present invention the host machine on which the application is being executed is not modified in any way by the process of decryption and installation since this is purely a run-time operation which modifies volatile RAM which only persists whilst the computer system is powered up. This is in direct contrast to Altberg in which the primary objective of the method of Altberg is to install an application onto a computer system so that the application remains on the computer system even after the computer system has been restarted or powered down.

Altberg provides a method for installing an application onto a computer system but does not provide any way of replacing missing required sub-routines during execution of the application as specified in the claims of the present application. Altberg thus suffers from the exact problems which the present invention intends to overcome, namely problems arising from conflicting versions of sub-routines or missing sub-routines during execution of an application.

It is therefore respectfully submitted that claims 27-28, 32, 35-38, 40, 42-44, 46 and 48-49 are not anticipated by Altberg.

The examiner contends that claims 29, 31, 39 and 50 are unpatentable over Altberg and Caron.

As discussed above, Altberg fails to disclose an executable which incorporates a decryption routine and Altberg fails to disclose a decryption routine that is operable during execution of the application.

Caron relates to "compiling computer programs written in a high level programming language into executable form" (column 1, lines 6-9). Caron is concerned with compiling executable applications and, like Altberg, is concerned purely with events which occur before a computer application is executed. Therefore, applicant submits that a

person of ordinary skill in the art would not consider the teachings of Caron when developing a system such as that of the present invention which is concerned with improving the reliability of an application during execution of the application.

Even if the person of ordinary skill in the art were to combine the teachings of Altberg with the teachings of Caron, the person of ordinary skill in the art would not reach the present invention because neither Altberg nor Caron disclose the provision of a decryption routine that is operable during execution of an application. Applicant submits that Altberg and Caron would not give a person of ordinary skill in the art any motivation whatsoever to reach the present invention because neither Altberg nor Caron disclose a) a single executable application or b) a decryption routine that is operable during execution of an application, with a view to increasing the stability of an application during execution.

It is therefore respectfully submitted that claims 29, 31, 39 and 50 are patentable.

The examiner contends that claims 33-34, 41 and 47 are unpatentable over Altberg and Shen.

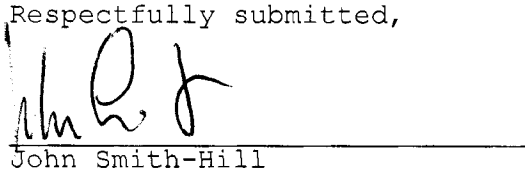
Shen discloses a "back-up/restore method and its control apparatus that can generate back-up files automatically for each file stored in a recording medium and enable easy restoration of the original file to a state of designated time period backing from the current time" (column 1, lines 6-10). Shen, like Altberg, is concerned with restoring files in a computer system to the correct predetermined positions in the system ready for when an application is executed. Shen is thus concerned only with restoring missing files before an application is executed. Applicant submits that a person of ordinary skill in the art would not combine the teachings of Altberg with the teachings of Shen when trying to develop a system such as the present invention which is concerned with operation during execution of an application because both Altberg and Shen are concerned with the installation or restoration of files before an application is executed.

Even if a person of ordinary skill in the art were to combine the

teachings of Altberg with the teachings of Shen then the person of ordinary skill in the art would not reach all of the features of the present invention because Altberg and Shen do not disclose an executable application that includes a decryption routine or a decryption routine that is operable during execution of an application.

It is therefore submitted that claims 33-34, 41 and 47 are patentable.

Respectfully submitted,



John Smith-Hill
Reg. No. 27,730

SMITH-HILL & BEDELL, P.C.
16100 N.W. Cornell Road, Suite 220
Beaverton, Oregon 97006

Tel. (503) 574-3100
Fax (503) 574-3197

Docket: FORR 2275